Lattice Signatures Without Trapdoors

Vadim Lyubashevsky[⋆]

INRIA / École Normale Supérieure

Abstract. We provide an alternative method for constructing lattice-based digital signatures which does not use the "hash-and-sign" methodology of Gentry, Peikert, and Vaikuntanathan (STOC 2008). Our resulting signature scheme is secure, in the random oracle model, based on the worst-case hardness of the $\tilde{O}(n^{1.5})$-SIVP problem in general lattices. The secret key, public key, and the signature size of our scheme are smaller than in all previous instantiations of the hash-and-sign signature, and our signing algorithm is also quite simple, requiring just a few matrix-vector multiplications and rejection samplings. We then also show that by slightly changing the parameters, one can get even more efficient signatures that are based on the hardness of the Learning With Errors problem. Our construction naturally transfers to the ring setting, where the size of the public and secret keys can be significantly shrunk, which results in the most practical to-date provably secure signature scheme based on lattices.

1 Introduction

The versatility of lattice-based cryptography has elevated it to the status of a promising potential alternative to cryptography based on standard security assumptions such as factoring and discrete log. But before lattices can become a viable replacement for number-theoretic schemes, it is crucial to have efficient lattice-based constructions of the most ubiquitous cryptographic primitives in practical applications, which are arguably encryption schemes and digital signatures.

On the encryption front, lattice-based schemes have been making a lot of progress with recent provably-secure schemes [Reg09,LPR10,LP11,SS11] being almost as practical as (and actually looking quite similar to) the deployed NTRU [HPS98] encryption scheme, which in turn has many advantages over number theory-based schemes. Lattice-based signatures, on the other hand, have been a different story. An early attempt at lattice-based signatures was the GGH scheme [GGH97] was completely broken in [NR09]. The NTRU signature scheme had an even more more tumultuous history since its introduction in 2001 [HPS01], with attacks [GS02] being followed by fixes [HHGP$^+$03], until its basic version was also completely broken by Nguyen and Regev [NR09].

Provably secure lattice-based signature schemes were finally constructed in 2008, when Gentry, Peikert, and Vaikuntanathan [GPV08] constructed a "hash-and-sign" signature scheme based on the hardness of worst-case lattice problems and Lyubashevsky and Micciancio [LM08] constructed a one-time signature based on the hardness of worst-case ideal lattice problems. The hash-and-sign signatures were rather inefficient (with signatures being megabytes long) and the one-time signature, while being relatively short, still required Merkle trees to become a full-fledged signature. Building on [LM08], Lyubashevsky proposed a digital signature, using the Fiat-Shamir framework [FS86] based on the hardness of ideal lattice problems [Lyu09]. This latter scheme has signature lengths on the order of 60000 bits for reasonable security parameters, and while closer to being practical, it is still not as small as one would like. Subsequently, lattice-based signature schemes without random oracles were also constructed [CHKP10,Boy10], but they are all much less efficient in practice than their random oracle-using counterparts.

[⋆] Work supported in part by the European Research Council.

1.1 Related Work and Our Results

A common thread running through constructions of digital signatures in the random oracle model, whether using the hash-and-sign or the Fiat-Shamir technique [FS86], is to force the distribution of the signature to be statistically independent of the secret key. If this property is achieved, then by programming the random oracle, one can hope to produce the valid signatures requested by the potential forger in the security reduction, without knowing the secret key. Then, when the forger produces a signature of a new message, it can be used to solve the underlying hard problem. In the case of lattices, the underlying hard problem is usually the Small Integer Solution (SIS) problem in which one is given a matrix $\mathbf{A}$ and is asked to find a *small* vector $\mathbf{v}$ such that $\mathbf{Av} = 0 \bmod q$. The length of $\mathbf{v}$ is very close to the length of signatures in the scheme, and thus the challenge for improving lattice-based signatures based on SIS is to reduce the norm of the signatures produced by the signing algorithm.

In lattice-based hash-and-sign signatures [GPV08], every signer has a personal uniformly random public matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and an associated secret "trapdoor" $\mathbf{S} \in \mathbb{Z}_q^{m\times m}$ with small coefficients such that $\mathbf{AS} = 0 \bmod q$. To sign a message μ, the signer uses his secret key $\mathbf{S}$ to produce a short signature vector $\mathbf{z}$, whose distribution is independent of $\mathbf{S}$, such that $\mathbf{Az} = \mathrm{H}(\mu) \bmod q$, where H is a cryptographic hash function. Since the length of $\mathbf{z}$ roughly depends on the norms of the columns of $\mathbf{S}$, improving the hash-and-sign signature scheme involves coming up with better algorithms for generating the pairs $(\mathbf{A}, \mathbf{S})$ such that $\mathbf{S}$ has smaller dimensions and smaller coefficients. Using the original algorithm due to Ajtai [Ajt99], the signature scheme of [GPV08] produced signatures of norm $\tilde{O}(n^{1.5})$. A subsequent improvement of the key-generation algorithm by Alwen and Peikert [AP11] lowered the signature length to $\tilde{O}(n)$, and the very recent algorithm of Micciancio and Peikert [MP12] further reduces the constants (and removes some logarithmic factors) from the previous algorithms.

There has been much less progress in the direction of building lattice-based signature schemes using the Fiat-Shamir technique. In fact, the only such scheme[1] is the ring-based one of Lyubashevsky [Lyu09], in which the signature vectors are of norm $\tilde{O}(n^{1.5})$. The first contribution of this current work is adapting the ring-SIS based scheme from [Lyu09] to one based on the hardness of the regular SIS problem which results in signatures of the same $\tilde{O}(n^{1.5})$ length[2]. Our second contribution is analogous to what the works [AP11,Pei10,MP12] did for hash-and-sign signatures – reduce the signature length to $\tilde{O}(n)$ (of course the issues that have to be dealt with are completely different). Our third contribution is showing that the parameters of our scheme can be set so that the resulting scheme produces much shorter signatures, but is now based on the hardness of the Learning With Errors (LWE) problem [Reg09] or on the hardness of a low-density version of the SIS problem. All our results very naturally carry over to the ring setting, where the key bit-size is reduced by a factor of approximately n (some sample parameters are given in Figure 2).

[1] We mention that the lattice-based identification schemes of Lyubashevsky [Lyu08a] and Kawachi et al. [KTX08], while may be converted into signature schemes, are inherently inefficient because every round of the ID scheme has soundness error at least 1/2.

[2] As a side note to this first result, we think that it is interesting to point out that the ring-structure, which seemed so native to [Lyu09] (and to [LM08]), turns out to not actually provide any additional functionality, with its purpose being only to shorten the key-sizes and make operations more efficient. This somewhat resembles the recent developments in constructions of fully-homomorphic encryption schemes, where the additional structure of ideal lattices was crucially used in earlier constructions [Gen09,Gen10,BV11b], but was subsequently shown to be unnecessary [BV11a,AFFP11].

Our signature scheme is also quite simple, requiring no pre-image sampling over arbitrary lattices. All we do is sample the Normal distribution over $\mathbb{Z}^m$, compute a vector-matrix product, do a random oracle query, compute another vector-matrix product (this time the vector is sparse), and rejection sample. In fact, in an online/offline setting where we can do pre-computations before being given the message to sign, the online phase simply consists of doing a few vector additions (since the matrix is being multiplied by a sparse vector) and rejection sampling.

1.2 Techniques

We now briefly sketch our signature scheme and describe the issues involved in lowering the size of the signature. The secret key is a matrix $\mathbf{S} \in \mathbb{Z}_q^{m \times k}$ with small coefficients, and the public key consists of the matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} = \mathbf{AS} \bmod q$. The matrix $\mathbf{A}$ can be shared among all users, but the matrix $\mathbf{T}$ is individual. To sign a message, the signer first picks a vector $\mathbf{y} \in \mathbb{Z}_q^m$ according to some distribution D. Then he computes $\mathbf{c} \in \mathbb{Z}_q^k$ where $\mathbf{c} \leftarrow \mathrm{H}(\mathbf{Ay} \bmod q, \mu)$, and computes the potential signature vector $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$ (there is no reduction modulo q in this step). The vector $\mathbf{z}$, along with $\mathbf{c}$, will then be output as the signature based on some criteria with the end goal being that the distribution of $(\mathbf{z}, \mathbf{c})$ should be independent of the secret key matrix $\mathbf{S}$.

Choosing when to output the pair $(\mathbf{z}, \mathbf{c})$ can be seen as a kind of *rejection sampling.* If f and g are probability distributions and $M \in \mathbb{R}$ is such that for all x, $f(x) \le Mg(x)$, then if one samples elements z from g and outputs them with probability $f(z)/(Mg(z))$, the resulting distribution is exactly f, and the expected amount of time needed to output a sample is M.

Our goal, in the signature scheme above, is to come up with distributions f and D so that for all $\mathbf{x}$, two properties are satisfied: there is a small constant M such that $f(\mathbf{x}) \le Mg(\mathbf{x})$, where g is the distribution generated by first picking $\mathbf{y}$ from D and adding it to $\mathbf{Sc}$ for some random $\mathbf{c}$; and the expected value of vectors distributed according to f (which is the length of the signature) is as small as possible. The idea in [Lyu09], when put into the above framework, was to choose $\mathbf{y}$ uniformly from an m-dimensional sphere[3] β_{r+v} of radius $r + v$, where r is some number and v is the maximum possible length of the vector $\mathbf{Sc}$, and only output $\mathbf{z}$ if it fell into a sphere β_r of radius r. It's not hard to check that if f is the uniform distribution over the sphere β_r, then by setting $M = vol(\beta_{r+v}/\beta_r) \approx (1 + v/r)^m$, the distribution of $\mathbf{z}$ is exactly f. But in order to keep M small, we need $r > mv = \tilde{\Theta}(m^{1.5}) = \tilde{\Theta}(n^{1.5})$, and so the vectors $\mathbf{z}$ have length $\tilde{O}(n^{1.5})$.

In our present work we show that we can do better by choosing f and D to be the m-dimensional Normal distribution with standard deviation $\sigma = \tilde{\Theta}(v) = \tilde{\Theta}(\sqrt{m})$, and only require that $f(\mathbf{x}) \le Mg(\mathbf{x})$ for the $\mathbf{x}$ that are not too big. We can then show that M can be set to a constant, and the rejection sampling algorithm produces a distribution that is statistically close to the distribution of f. This means that the expected value of the length of the signature of $\mathbf{z}$ is $\sigma\sqrt{m} = \tilde{O}(m) = \tilde{O}(n)$. We prove the technical rejection sampling theorem in Section 4 and then prove the security of the above signature scheme based on the hardness of the SIS problem in Section 5.

Notice that the length of the signature is greatly affected by the parameter m, and lowering m, while leaving everything else the same would produce even shorter signatures. The danger of doing this is that the problem of recovering $\mathbf{S}$ when given $\mathbf{A}$ and $\mathbf{AS} \bmod q$ now becomes easier (and is no longer based on the SIS problem). The intuition is then to set all the parameters so that the hardness of recovering the secret key is equal, in practice, to the hardness of forging a signature. In Section 6 we explain how the parameters can be significantly lowered by making our scheme be based on the LWE problem instead of on SIS.

[3] In [Lyu09], it was actually a box, but it does not make a difference for the analysis here.

1.3 A Comparison with Hash-and-Sign Signatures

On the theoretical side, both the scheme constructed in this paper and the hash-and-sign scheme that uses the trapdoor sampling algorithms of [AP11,Pei10] are based on the hardness of finding a vector of length $\tilde{O}(n)$ in SIS instances, which by the worst-case to average-case reduction of Micciancio and Regev [MR07] is as hard as solving approximate SIVP with a factor of $\tilde{O}(n^{1.5})$ in all n-dimensional lattices. On the practical side, however, the bit-length of our signature and keys (see Figure 2) are approximately two orders of magnitude smaller for the same security level (see [RS10] and also Figure 2 in [MP12]). This is mostly due to the constants that are hidden in the big-Oh notation of the trapdoor generation algorithms of [AP11] and [Pei10].

As mentioned earlier, in a concurrent and independent work, Micciancio and Peikert greatly improved the constants, and in some cases even removed some logarithmic factors, in the trapdoor sampling algorithms [MP12]. While the proof techniques are completely different, there are some high-level similarities between the two schemes. The public key in our scheme is $(\mathbf{A}, \mathbf{AS})$ where $\mathbf{A}$ is a random matrix mod q and $\mathbf{S}$ is a secret matrix with small coefficients. In [MP12], the public key is $(\mathbf{A}, \mathbf{AS} + \mathbf{G})$ where $\mathbf{G}$ is an additional public matrix with a very "simple" form. In our scheme, the signature of a message is an ordered pair $(\mathbf{Sc} + \mathbf{y}, \mathbf{c})$ where $\mathbf{c}$ is a function (that invokes a random oracle) of the message and the vector $\mathbf{y}$ is there to "hide" the shift $\mathbf{Sc}$; while in [MP12], the signature is $(\mathbf{Sc} + \mathbf{y}_1, \mathbf{c} + \mathbf{y}_2)$ where $\mathbf{c}$ is a (different, random oracle-invoking) function of the message and $\mathbf{y}_i$ also serve the purpose of hiding the shift $\mathbf{Sc}$ (and $\mathbf{c}$ itself). While the schemes may look similar, under the surface they behave rather differently.

The most interesting and significant difference occurs in the way the signatures are generated. In our scheme, the vector $\mathbf{c}$ is a very sparse $-1/0/1$ vector whose entropy is as small as the security parameter, but we *must* output it as part of the signature. In [MP12], however, the size of the elements in $\mathbf{c}$ depends *inversely* on the number of columns of $\mathbf{S}$, but one only outputs a *perturbed* version of $\mathbf{c}$ as part of the signature. Notice that the size of our signature is therefore dominated by the number of rows of $\mathbf{S}$ multiplied by the number of bits needed to represent elements in the vector $\mathbf{Sc} + \mathbf{y}$, whereas in [MP12], the number of columns of $\mathbf{S}$ may also play a significant role in the signature length.

The advantage in [MP12] due to the fact that $\mathbf{c}$ is never output in the clear is that they may tailor the perturbations $\mathbf{y}_1, \mathbf{y}_2$ to the particular $\mathbf{S}$ that they are supposed to hide, which allows these perturbations to be smaller than ours in the case that $\mathbf{S}$ has enough columns to allow $\mathbf{c}$ to be "small enough". When instantiating both signature schemes based on the worst-case hardness of the SIS problem, $\mathbf{S}$ needs to have a large number of rows, and thus the fact that the bit-size of the entries of the signature from [MP12] is smaller than of those in our scheme, may make the scheme from [MP12] more compact. On the other hand, if one is to instantiate the more practical version of the schemes based on the hardness of the LWE problem, then the number of rows in $\mathbf{S}$ could be significantly smaller, and thus the fact that the size of our signature does not depend on the number of columns of $\mathbf{S}$ gives it an advantage over the one in [MP12]. We direct the reader to our sample instantiations in Figure 2 where one can see the signature size rapidly decreasing as the number of rows (denoted by m) shrinks. The trade-off is that as the number of rows shrinks, the worst-case hardness assumption becomes stronger, but it is still believed that the security of the average-case problem remains the same (see Section 3).

Additionally, the number of columns in our secret key $\mathbf{S}$ needs to only be large enough to support multiplication by $\mathbf{c}$, which allows the number of columns to be significantly smaller than in the secret key of [MP12], where, for technical reasons, reducing the number of columns of $\mathbf{S}$ ends up increasing the coefficients of $\mathbf{c}$, and thus possibly increasing the size of the signature. This allows our secret key to be smaller that the one in [MP12]. Compared to the one concrete instantiation (based on the hardness of the SIS problem) provided in [MP12], where the key size is approximately $2^{26.5}$ bits and the signature is a 13800 dimensional vector of length 92000, thus requiring at least $13800 \cdot \log(92000/\sqrt{13800}) \approx 130000$ bits to represent, for the same security level, some of our instantiations have the signature bit-length about 25% longer, with the benefit of having the keys be about 10 times smaller (column I of Figure 2). For different instantiations, we can have the signature bit-length be about 45% shorter and have the same key size (column III of Figure 2).

1.4 Organization of the Paper

In Section 3, we review the average-case SIS problem and its variants upon which the security of our signature is based. In Section 4, we review some facts about the Normal distribution and prove a rejection sampling theorem that will be used for proving that the distribution of our signature output is statistically indistinguishable of the secret key. In Section 5, we construct a signature secure based on the hardness of SIS, and in Section 6, we modify it to be more efficient, but now based on LWE or low-density SIS. In Section 7, we sketch how to transfer our signature into the ring setting. For simplicity, we do not introduce rings or LWE until the sections in which they are first used for signature scheme constructions.

2 Preliminaries

2.1 Notation

Throughout the paper, we will assume that q is a small (i.e. polynomial-size) prime number and elements in $\mathbb{Z}_q$ are represented by integers in the range $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$. We will represent vectors by bold-face letters, and matrices by bold-face capital letters. We will assume that all vectors are column vectors, and $\mathbf{v}^T$ will denote the transpose of the vector $\mathbf{v}$. The ℓ_p norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_p$, and we will usually avoid writing the p for the ℓ_2 norm. Whenever dealing with elements that are in $\mathbb{Z}_q$, we always explicitly assume that all operations in which they are involved end with a reduction modulo q. Thus for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ and a vector $\mathbf{s} \in \mathbb{Z}^n$, the product $\mathbf{As}$ is a vector in $\mathbb{Z}_q^n$. For a distribution $\mathcal{D}$, we use the notation $x \overset{\$}{\leftarrow} \mathcal{D}$ to mean that x is chosen according to the distribution $\mathcal{D}$. If S is a set, then $x \overset{\$}{\leftarrow} S$ means that x is chosen uniformly at random from S. For an event E, we write $Pr[E; x_1 \overset{\$}{\leftarrow} \mathcal{D}_1, \ldots, x_k \overset{\$}{\leftarrow} \mathcal{D}_k]$ to mean the probability that E occurs when the x_i are chosen from distributions $\mathcal{D}_i$. All logarithms are base 2.

2.2 Digital Signatures

We recall the definitions of signature schemes and what it means for a signature scheme to be secure.

Definition 2.1. *A* signature scheme *consists of a triplet of polynomial-time (possibly probabilistic) algorithms (G, S, V) such that for every pair of outputs (s, v) of $G(1^n)$ and any n-bit message m,*

$$Pr[V(v, m, S(s, m)) = 1] = 1$$

where the probability is taken over the randomness of algorithms S and V.

In the above definition, G is called the key-generation algorithm, S is the signing algorithm, V is the verification algorithm, and s and v are, respectively, the signing and verification keys.

A signature scheme is said to be secure if there is only a negligible probability that any forger, after seeing signatures of messages of his choosing, can sign a message whose signature he has not already seen [GMR88].

Definition 2.2. *A signature scheme (G, S, V) is said to be secure if for every polynomial-time (possibly randomized) forger $\mathcal{F}$, the probability that after seeing the public key and $\{(\mu_1, S(s, \mu_1)), \ldots, (\mu_q, S(s, \mu_q))\}$ for any q messages μ_i of its choosing (where q is polynomial in n), $\mathcal{F}$ can produce $(\mu \neq \mu_i, \sigma)$ such that $V(v, \mu, \sigma) = 1$, is negligibly small. The probability is taken over the randomness of G, S, V, and $\mathcal{F}$.*

In the standard security definition of a signature scheme, the forger should not be able to produce a signature of a new message. A stronger notion of security, called *strong unforgeability* requires that in addition to the above, a forger shouldn't even be able to come up with a different signature for a message whose signature he has already seen. The schemes presented in this paper satisfy this stronger notion of unforgeability.

3 The SIS Problem and its Variants

In this section, we will define the average-case problems upon whose security our signature schemes will be based. All these problems fall into the category of the Small Integer Solution (SIS) problem, which is essentially the knapsack problem over elements in $\mathbb{Z}_q^n$.

Definition 3.1 **(ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem).** *Given a random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ find a vector $\mathbf{v} \in \mathbb{Z}^m \setminus \{0\}$ such that $\mathbf{A}\mathbf{v} = 0$ and $\|\mathbf{v}\| \leq \beta$.*

In order for the above problem to not be vacuously hard, we need to have $\beta \geq \sqrt{m}q^{n/m}$ in order for there to exist a solution $\mathbf{v}$. The signature scheme that we construct in Section 5 is based on the presumed hardness of the above problem. In Section 6, we construct a more efficient signature scheme based on the hardness of SIS variants defined below.

Definition 3.2 **($\mathrm{SIS}_{q,n,m,d}$ distribution).** *Choose a random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{s} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^m$ and output $(\mathbf{A}, \mathbf{A}\mathbf{s})$.*

Definition 3.3 **($\mathrm{SIS}_{q,n,m,d}$ search problem).** *Given a pair $(\mathbf{A}, \mathbf{t})$ from the $\mathrm{SIS}_{q,n,m,d}$ distribution, find a $\mathbf{s} \in \{-d, \ldots, 0, \ldots, d\}^m$ such that $\mathbf{A}\mathbf{s} = \mathbf{t}$.*

Definition 3.4 **($\mathrm{SIS}_{q,n,m,d}$ decision problem).** *Given a pair $(\mathbf{A}, \mathbf{t})$ decide, with non-negligible advantage, whether it came from the $\mathrm{SIS}_{q,n,m,d}$ distribution or whether it was generated uniformly at random from $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$.*

Depending on the relationship between its parameters, the $\mathrm{SIS}_{q,n,m,d}$ search (and decision) problem has somewhat different characteristics. If, for example, we have $d \ll q^{n/m}$, then with very high probability there is only one vector $\mathbf{s}$ whose coefficients have absolute value at most d such that $\mathbf{A}\mathbf{s} = \mathbf{t}$, and such instances of the $\mathrm{SIS}_{q,n,m,d}$ problem are said to be *low-density* instances (borrowing from terminology used to describe instances of the random subset sum problem). On the other hand, if $d \gg q^{n/m}$ then the $\mathrm{SIS}_{q,n,m,d}$ distribution is actually statistically close to uniform over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ (by the leftover hash lemma) and there are many possible solutions $\mathbf{s}$ for which $\mathbf{As} = \mathbf{t}$. These instances are traditionally called *high-density* instances. We will discuss the hardness of the SIS problem in Section 3.2, below, but we will now mention that the hardest instances are those in which $d \approx q^{n/m}$.

Notice that if $m \geq 2n$, then the matrix $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$ will, with high probability, contain n columns that are linearly independent over $\mathbb{Z}_q$ (when $m \geq 2n$ and q is a prime of size at least $2m$, this will be true with probability $e^{-\Omega(n)}$). Without loss of generality, assume that the last n columns of $\mathbf{A}$ are linearly independent, and so $\mathbf{A} = [\mathbf{A}_1||\mathbf{A}_2]$ where $\mathbf{A}_2$ is an $n \times n$ invertible matrix. If we consider the matrix $\mathbf{A}' = \mathbf{A}_2^{-1}\mathbf{A} = [\mathbf{A}_2^{-1}\mathbf{A}_1||\mathbf{I}]$, where $\mathbf{I}$ is an $n \times n$ identity matrix, then we have $\mathbf{Av} = 0$ iff $\mathbf{A}'\mathbf{v} = 0$, and so the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem is equally hard if the last n columns of the matrix $\mathbf{A}$ form the identity matrix. Similarly, given an instance $(\mathbf{A}, \mathbf{t})$ of the $\mathrm{SIS}_{q,n,m,d}$ problem, we can change it to $(\mathbf{A}_2^{-1}\mathbf{A}, \mathbf{A}_2^{-1}\mathbf{t})$, and a solution for one will be exactly the same as the solution for the other. Therefore throughout this paper we will assume, without loss of generality, that the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is of the form $\mathbf{A} = [\bar{\mathbf{A}}||\mathbf{I}]$, where $\bar{\mathbf{A}}$ is uniformly generated in $\mathbb{Z}_q^{n\times(m-n)}$. For reasons related to lattices, when $\mathbf{A}$ is in this form, we will refer to it as being in *Hermite Normal Form* [MR08].

3.1 Relations Between the SIS Variants

We now state some results about the relationship between the SIS variants defined above. The first relationship is an adaptation of a classic theorem of Impagliazzo and Naor [IN96], who showed that the decisional version of the random subset sum problem is as hard as the search version. This theorem has been recently generalized by Micciancio and Mol [MM11].

Theorem 3.5. [IN96,MM11] *If d is polynomial in n, then there is a polynomial-time reduction from the $\mathrm{SIS}_{q,n,m,d}$ search problem to the $\mathrm{SIS}_{q,n,m,d}$ decision problem.*

The next lemma shows that the *decision* $\mathrm{SIS}_{q,n,m,d}$ problem gets harder when the value of d increases. This is a rather intuitive result since the decision $\mathrm{SIS}_{q,n,m,d}$ problem becomes vacuously hard when $d \gg q^{n/m}$ since the $\mathrm{SIS}_{q,n,m,d}$ distribution will be statistically close to uniform.

Lemma 3.6. *For any non-negative integer α such that $gcd(2\alpha + 1, q) = 1$, there is a polynomial-time reduction from the $\mathrm{SIS}_{q,n,m,d}$ decision problem to the $\mathrm{SIS}_{q,n,m,(2\alpha+1)d+\alpha}$ decision problem.*

Proof. To prove the lemma, we will show a transformation that maps the $\mathrm{SIS}_{q,n,m,d}$ distribution to the $\mathrm{SIS}_{q,n,m,(2\alpha+1)d+\alpha}$ distribution, and maps the uniform distribution over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ to itself. Given $(\mathbf{A}, \mathbf{t})$, create a random vector $\mathbf{r} \overset{\$}{\leftarrow} \{-\alpha, \ldots, 0, \ldots, \alpha\}^m$ and output $(\mathbf{A}, (2\alpha + 1)\mathbf{t} + \mathbf{Ar})$. First observe that because $2\alpha + 1$ is relatively prime to q, our transformation maps the uniform distribution to itself. And if $(\mathbf{A}, \mathbf{t})$ came from the $\mathrm{SIS}_{q,n,m,d}$ distribution, then $(2\alpha + 1)\mathbf{t} + \mathbf{Ar} = \mathbf{A}((2\alpha + 1)\mathbf{s} + \mathbf{r})$, and since $\mathbf{s}$ was chosen uniformly at random from $\{-d, \ldots, 0, \ldots, d\}^m$, it's not hard to see that $(2\alpha+1)\mathbf{s}+\mathbf{r}$ is uniformly random in $\{-(2\alpha+1)d-\alpha, \ldots, 0, \ldots, (2\alpha+1)d+\alpha\}^m$. □

We now show that if $m = 2n$ and one can solve the can solve ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem for a small-enough β, then one can solve the decision $\mathrm{SIS}_{q,n,m,d}$ problem. This result is essentially folklore (see [MR08]), but we prove it here for completeness.

Lemma 3.7. *If $m = 2n$ and $4d\beta\sqrt{m} \le q$, then there is a polynomial-time reduction from solving the* $\mathrm{SIS}_{q,n,m,d}$ *decision problem to the* ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ *problem.*

Proof. Given an instance $(\mathbf{A}, \mathbf{t})$ of the $\mathrm{SIS}_{q,n,m,d}$ decision problem where $\mathbf{A} = [\bar{\mathbf{A}}||\mathbf{I}]$ is in Hermite Normal Form (where $\bar{\mathbf{A}}$ and $\mathbf{I}$ are both $n \times n$ square matrices), use the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ oracle on the matrix $\mathbf{A}' = [\bar{\mathbf{A}}^T||\mathbf{I}]$ to find a vector $\mathbf{v} = [(\mathbf{v}_1)^T||(\mathbf{v}_2)^T]^T$ such that $\mathbf{A}'\mathbf{v} = \bar{\mathbf{A}}^T\mathbf{v}_1 + \mathbf{v}_2 = 0$. Now consider the inner product $\langle \mathbf{v}_1, \mathbf{t}\rangle = \mathbf{v}_1^T\mathbf{t}$. If $\mathbf{t} = \mathbf{A}\mathbf{s} = \bar{\mathbf{A}}\mathbf{s}_1 + \mathbf{s}_2$, then

$$\mathbf{v}_1^T\mathbf{t} = \mathbf{v}_1^T\bar{\mathbf{A}}\mathbf{s}_1 + \mathbf{v}_1^T\mathbf{s}_2 = -\mathbf{v}_2^T\mathbf{s}_1 + \mathbf{v}_1^T\mathbf{s}_2 \tag{1}$$

and since $\|\mathbf{v}\| \le \beta$ and all the coefficients of $\mathbf{s}$ are at most d, we have that $|\langle \mathbf{v}_1, \mathbf{t}\rangle| \le \beta d\sqrt{m} \le q/4$. On the other hand, if $\mathbf{t}$ is uniformly random, then $\langle \mathbf{v}_1, \mathbf{t}\rangle$ will also be uniformly random in $\mathbb{Z}_q$. Therefore the distinguisher for the $\mathrm{SIS}_{q,n,m,d}$ decision problem simply looks at the absolute value of the inner product of $\mathbf{v}_1$ and $\mathbf{t}$ and says that $(\mathbf{A}, \mathbf{t})$ came from the $\mathrm{SIS}_{q,n,m,d}$ distribution if the absolute value is at most $q/4$, and he says that $(\mathbf{A}, \mathbf{t})$ is uniform, otherwise. In the case that $(\mathbf{A}, \mathbf{t})$ comes from the $\mathrm{SIS}_{q,n,m,d}$ distribution, the distinguisher will always be correct, and in the case of the uniform distribution, he will make an error with probability 1/2. □

In the above lemma, the parameters were set such that the distinguisher only has one-sided error, but it is actually possible to be looser with the bound for $d\beta/q$ and still be able to distinguish with non-negligible probability. In fact, solving the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem is the most efficient method known for solving the decisional $\mathrm{SIS}_{q,n,m,d}$ problem. We will now outline the basic idea, and refer the reader to [MR08] for more details. The norm of the m-dimensional vector $[\mathbf{s}_1^T||\mathbf{s}_2^T]$ is concentrated tightly around $\sqrt{d(d+1)m/3}$, which is the same as the norm of an m-dimensional normal variable with standard deviation $\psi = \sqrt{d(d+1)/3}$ (see Section 4 for a discussion about the normal distribution). Thus, heuristically, the distribution $-\mathbf{v}_2^T\mathbf{s}_1 + \mathbf{v}_1^T\mathbf{s}_2$ in Equation (1) will be distributed as a 1-dimensional (discrete) normal variable with standard deviation $\psi\beta$. It was shown in [MR07, Lemma 3.3 and Lemma 4.1] that if $\psi\beta\sqrt{2\pi}/q > \sqrt{\ln(1/\epsilon)/\pi}$, then a 1-dimensional normal variable with standard deviation $\psi\beta$ is approximately within statistical distance ϵ of the uniform distribution over q. If we want the decision $\mathrm{SIS}_{q,n,m,d}$ problem to be hard, then we should make sure that the preceding equality is satisfied for $\epsilon \approx 2^{-100}$. Thus we will be aiming for

$$\beta\psi/q \ge 2, \text{ where } \psi = \sqrt{d(d+1)m/3}. \tag{2}$$

3.2 Computational Hardness of ℓ_2-SIS

The SIS problem gained prominence when Ajtai showed that solving its random, high-density instances is as hard as solving worst-case instances of certain lattice problems [Ajt96]. Ajtai's connection between SIS and worst-case lattice problems has subsequently been tightened up to the currently best result of Micciancio and Regev [MR07], who show that (for a large-enough q) solving random instances of the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem is as hard as solving the $\tilde{O}(\sqrt{n}\beta)$-SIVP problem in all lattices of dimension n. While these seminal results give us a lot of confidence in the hardness of SIS, they are not very useful for guiding us in parameter selection when building cryptographic primitives, mainly because solving the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem requires one to solve lattice problems in a dimension somewhere between n and m [GN08,MR08], whereas the hardness of the worst-case $\tilde{O}(\sqrt{n}\beta)$-SIVP problem is for lattices of dimension only n – which is a seemingly much easier problem. For this reason, parameter choices for lattice-based primitives have been mostly proposed based on the hardness of average-case instances of the SIS (and Ring-SIS[4]) problem. For example, the SWIFFT collision-resistant hash function [LMPR08] is essentially based on the hardness of the ring version of the ℓ_2-$\mathrm{SIS}_{257,64,1024,32}$ problem, which under the worst-case to average-case reduction would only make it as hard as lattice problems in dimension $n = 64$ (lattice problems in such a small dimension are easy). Nevertheless, the underlying average-case SIS problem seems to be much harder, and as far as we are aware, no progress has been made towards lowering the claimed 2^{106} time for finding a collision using birthday attacks.

The computational hardness of knapsack problems, to which SIS belongs, has been studied since the early 1980's, and the main technique for solving random instances of the knapsack problem has been lattice reduction [LO83]. The basic idea is to define the lattice $\mathcal{L}(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A}\mathbf{v} = 0\}$ and then use lattice reduction algorithms to find short vectors in $\mathcal{L}(\mathbf{A})$. The experiments of Gama and Nguyen [GN08] showed that lattice-reduction algorithms are able to find vectors of length $\beta \le \delta^m \cdot det(\mathcal{L})^{1/m}$ in m-dimensional lattices $\mathcal{L}(\mathbf{A})$, where δ is a parameter that depends on the quality of the lattice-reduction algorithm being used. The factor δ of the currently best algorithms is around 1.01, and it is conjectured that a factor of 1.007 may be outside our reach for the foreseeable future [CN11]. In this paper, we use the value $\delta = 1.007$ for setting the parameters.

Using the results of [GN08], Micciancio and Regev [MR08] deduced that to solve high-density SIS instances, one should only use a maximum of $\sqrt{n \log q / \log \delta}$ of the m columns of the matrix $\mathbf{A}$ in the lattice-reduction algorithm, which should allow one to find a non-zero vector $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = 0$ of length

$$\min\left(q, 2^{2\sqrt{n \log q \log \delta}}\right). \tag{3}$$

4 Rejection Sampling and the Normal Distribution

Definition 4.1. *The continuous Normal distribution over $\mathbb{R}^m$ centered at $\mathbf{v}$ with standard deviation σ is defined by the function $\rho^m_{\mathbf{v},\sigma}(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$*

When $\mathbf{v} = 0$, we will just write $\rho^m_\sigma(\mathbf{x})$. We will define the *discrete* Normal distribution over $\mathbb{Z}^m$ as follows:

Definition 4.2. *The discrete Normal distribution over $\mathbb{Z}^m$ centered at some $\mathbf{v} \in \mathbb{Z}^m$ with standard deviation σ is defined as $D^m_{\mathbf{v},\sigma}(\mathbf{x}) = \rho^m_{\mathbf{v},\sigma}(\mathbf{x})/\rho^m_\sigma(\mathbb{Z}^m)$.*

In the above definition, the quantity $\rho^m_\sigma(\mathbb{Z}^m) = \sum\limits_{\mathbf{z}\in\mathbb{Z}^m} \rho^m_\sigma(\mathbf{z})$ is just a scaling quantity needed to make the function into a probability distribution. Also note that for all $\mathbf{v} \in \mathbb{Z}^m$, $\rho^m_{\mathbf{v},\sigma}(\mathbb{Z}^m) = \rho^m_\sigma(\mathbb{Z}^m)$, thus the scaling factor is the same for all $\mathbf{v}$.

Before stating the main theorem of this section, we will prove several facts about the discrete Normal distribution over $\mathbb{Z}^m$. The first lemma bounds the inner product of a discrete normal variable with any vector in $\mathbb{R}^m$.

Lemma 4.3. *For any vector $\mathbf{v} \in \mathbb{R}^m$ and any $\sigma, r > 0$,*

$$\Pr[|\langle \mathbf{z}, \mathbf{v}\rangle| > r; \mathbf{z} \overset{\$}{\leftarrow} D^m_\sigma] \le 2e^{-\frac{r^2}{2\|\mathbf{v}\|^2\sigma^2}}.$$

[4] See Section 7 for discussions about the ring versions of lattice problems.

Proof. For any $t > 0$, we have:

$$\begin{aligned}
E\left[\exp\left(\frac{t}{\sigma^2}\langle \mathbf{z}, \mathbf{v}\rangle\right)\right] &= \sum_{\mathbf{z}\in\mathbb{Z}^m} \Pr[\mathbf{z}] \exp\left(\frac{1}{\sigma^2}\langle \mathbf{z}, t\mathbf{v}\rangle\right) \\
&= \left(\sum_{\mathbf{y}\in\mathbb{Z}^m} \exp\left(\frac{-\|\mathbf{y}\|^2}{2\sigma^2}\right)\right)^{-1} \sum_{\mathbf{z}\in\mathbb{Z}^m} \exp\left(\frac{-\|\mathbf{z}\|^2}{2\sigma^2}\right) \exp\left(\frac{1}{\sigma^2}\langle \mathbf{z}, t\mathbf{v}\rangle\right) \\
&= \left(\sum_{\mathbf{y}\in\mathbb{Z}^m} \exp\left(\frac{-\|\mathbf{y}\|^2}{2\sigma^2}\right)\right)^{-1} \sum_{\mathbf{z}\in\mathbb{Z}^m} \exp\left(\frac{-\|\mathbf{z}-t\mathbf{v}\|^2}{2\sigma^2}\right) \exp\left(\frac{t^2\|\mathbf{v}\|^2}{2\sigma^2}\right) \\
&= \frac{\rho^m_{t\mathbf{v},\sigma}(\mathbb{Z}^m)}{\rho^m_\sigma(\mathbb{Z}^m)} \exp\left(\frac{t^2\|\mathbf{v}\|^2}{2\sigma^2}\right) \\
&\le \exp\left(\frac{t^2\|\mathbf{v}\|^2}{2\sigma^2}\right),
\end{aligned}$$

where the last inequality follows from [MR07, Lemma 2.9]. We now proceed to prove the claim of the lemma by applying Markov's inequality and then the above result. In particular, for any $t > 0$, we have:

$$\begin{aligned}
\Pr[\langle \mathbf{z}, \mathbf{v}\rangle > r] &= \Pr\left[\exp\left(\frac{t}{\sigma^2}\langle \mathbf{z}, \mathbf{v}\rangle\right) > \exp\left(\frac{tr}{\sigma^2}\right)\right] \\
&\le \frac{E\left[\exp\left(\frac{t}{\sigma^2}\langle \mathbf{z}, \mathbf{v}\rangle\right)\right]}{\exp\left(\frac{tr}{\sigma^2}\right)} \\
&\le \exp\left(\frac{t^2\|\mathbf{v}\|^2}{2\sigma^2} - \frac{tr}{\sigma^2}\right) \\
&\le \exp\left(-\frac{r^2}{2\|\mathbf{v}\|^2\sigma^2}\right),
\end{aligned}$$

where the last inequality comes from optimally setting $t = r/\|\mathbf{v}\|^2$. Since the distribution of $\mathbf{z}$ is symmetric around the origin, we also have that $\Pr[\langle \mathbf{z}, \mathbf{v}\rangle < -r] \le \exp\left(-\frac{r^2}{2\|\mathbf{v}\|^2\sigma^2}\right)$, and applying the union bound to the two inequalities gives us the claim in the lemma. □

Lemma 4.4.

1. *For any $k > 0$, $Pr[|z| > k\sigma; z \xleftarrow{\$} D^1_\sigma] \le 2e^{\frac{-k^2}{2}}$,*
2. *For any $\mathbf{z} \in \mathbb{Z}^m$, and $\sigma \ge 3/\sqrt{2\pi}$, $D^m_\sigma(\mathbf{z}) \le 2^{-m}$*
3. *For any $k > 1$, $Pr[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} D^m_\sigma] < k^m e^{\frac{m}{2}(1-k^2)}$.*

Proof. Item 1 follows directly from Lemma 4.3 by substituting $m = 1, r = k\sigma$, and $\mathbf{v} = 1$. To prove item 2, we write

$$D^m_\sigma(\mathbf{z}) = \frac{e^{-\|\mathbf{z}\|^2/(2\sigma^2)}}{\sum\limits_{\mathbf{x}\in\mathbb{Z}^m} e^{-\|\mathbf{x}\|^2/(2\sigma^2)}} \le \frac{1}{\sum\limits_{\mathbf{x}\in\mathbb{Z}^m} e^{-\|\mathbf{x}\|^2/(2\sigma^2)}} = \frac{1}{\left(\sum\limits_{x_1\in\mathbb{Z}} e^{-x_1^2/(2\sigma^2)}\right)\cdots\left(\sum\limits_{x_m\in\mathbb{Z}} e^{-x_m^2/(2\sigma^2)}\right)}.$$

We now use the fact that if $f(x)$ is a non-increasing function between $x = 0$ and infinity, we have $\sum_{x=0}^{\infty} f(x) \geq \int_0^{\infty} f(x)dx$. If we let $f(x) = e^{-x^2/(2\sigma^2)}$, then we have

$$\sum_{x\in\mathbb{Z}} f(x) = 2\sum_{x=0}^{\infty} f(x) - f(0) \geq 2\int\limits_0^{\infty} f(x)dx - f(0) = \int\limits_{-\infty}^{\infty} f(x)dx - f(0) = \sqrt{2\pi}\sigma - 1.$$

Therefore if $\sigma \geq 3/\sqrt{2\pi}$, we have $D_\sigma^m(\mathbf{z}) \leq 2^{-m}$.

For 3, we use the result of [Ban93, Lemma 1.5] which shows that for all lattices $\Lambda \in \mathbb{R}^m$ and constants $c \geq 1/\sqrt{2\pi}$,

$$\sum_{\mathbf{z}\in\Lambda,\|\mathbf{z}\|>c\sqrt{m}} e^{-\pi\|\mathbf{z}\|^2} < \left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^m \sum_{\mathbf{z}\in\Lambda} e^{-\pi\|\mathbf{z}\|^2}.$$

By scaling the lattice Λ by a factor of $1/s$, for any constant s, the above implies that for all s,

$$\sum_{\mathbf{z}\in\Lambda,\|\mathbf{z}\|>cs\sqrt{m}} e^{-\pi\|\mathbf{z}\|^2/s^2} < \left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^m \sum_{\mathbf{z}\in\Lambda} e^{-\pi\|\mathbf{z}\|^2/s^2}.$$

Setting $\Lambda = \mathbb{Z}^m$ and $s = \sqrt{2\pi}\sigma$, we obtain

$$Pr[\|\mathbf{z}\| > c\sqrt{2\pi}\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} D_\sigma^m] < \left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^m.$$

Finally, we set $c = k/\sqrt{2\pi}$. □

The last lemma that we prove will be instrumental in bounding the success probability of our rejection sampling algorithm.

Lemma 4.5. *For any* $\mathbf{v} \in \mathbb{Z}^m$*, if* $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$*, then*

$$Pr[D_\sigma^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z}) = O(1); \mathbf{z} \xleftarrow{\$} D_\sigma^m] = 1 - 2^{-\omega(\log m)},$$

and more specifically, for any $\mathbf{v} \in \mathbb{Z}^m$*, if* $\sigma = \alpha\|\mathbf{v}\|$ *for any positive* α*, then*

$$Pr[D_\sigma^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z}) < e^{12/\alpha + 1/(2\alpha^2)}; \mathbf{z} \xleftarrow{\$} D_\sigma^m] > 1 - 2^{-100}.$$

Proof. By definition, we have

$$D_\sigma^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z}) = \rho_\sigma^m(\mathbf{z})/\rho_{\mathbf{v},\sigma}^m(\mathbf{z}) = \frac{\exp\left(-\frac{\|\mathbf{z}\|^2}{2\sigma^2}\right)}{\exp\left(-\frac{\|\mathbf{z}-\mathbf{v}\|^2}{2\sigma^2}\right)} = \exp\left(\frac{-2\langle\mathbf{z},\mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\right).$$

Lemma 4.3 tells us that $|\langle\mathbf{z},\mathbf{v}\rangle|$ is smaller than $\omega(\sqrt{\log m}\|\mathbf{v}\|\sigma)$ with probability at least $1 - 2^{-\omega(\log m)}$. Thus with this same probability we have that

$$\exp\left(\frac{-2\langle\mathbf{z},\mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\right) < \exp\left(\frac{\omega(\sqrt{\log m}\|\mathbf{v}\|\sigma) + \|\mathbf{v}\|^2}{2\sigma^2}\right) = O(1),$$

where the last equality uses $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$.

More specifically, from Lemma 4.3, we know that $|\langle \mathbf{z}, \mathbf{v}\rangle|$ is smaller than $12\|\mathbf{v}\|\sigma$ with probability at least $1 - 2^{-100}$, and therefore we have that with probability at least $1 - 2^{-100}$,

$$\exp\left(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\right) < \exp\left(\frac{24\|\mathbf{v}\|\sigma + \|\mathbf{v}\|^2}{2\sigma^2}\right) = e^{12/\alpha + 1/(2\alpha^2)},$$

where the last equality uses $\sigma = \alpha\|\mathbf{v}\|$. □

We now prove the main theorem of this section.

Theorem 4.6. *Let V be a subset of $\mathbb{Z}^m$ in which all elements have norms less than T, σ be some element in $\mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists a constant $M = O(1)$ such that the distribution of the following algorithm $\mathcal{A}$:*

```
$\mathbf{v} \overset{\$}{\leftarrow} h$
$\mathbf{z} \overset{\$}{\leftarrow} D^m_{\mathbf{v},\sigma}$
output $(\mathbf{z}, \mathbf{v})$ with probability $\min\left(\frac{D^m_\sigma(\mathbf{z})}{MD^m_{\mathbf{v},\sigma}(\mathbf{z})}, 1\right)$
```

is within statistical distance $\frac{2^{-\omega(\log m)}}{M}$ of the distribution of the following algorithm $\mathcal{F}$:

```
$\mathbf{v} \overset{\$}{\leftarrow} h$
$\mathbf{z} \overset{\$}{\leftarrow} D^m_\sigma$
output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$
```

Moreover, the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$.

More concretely, if $\sigma = \alpha T$ for any positive α, then $M = e^{12/\alpha + 1/(2\alpha^2)}$, the output of algorithm $\mathcal{A}$ is within statistical distance $\frac{2^{-100}}{M}$ of the output of $\mathcal{F}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-100}}{M}$.

Proof. The proof of this theorem will follow from Lemmas 4.5 and a general "rejection sampling" lemma that we will now prove.

Lemma 4.7. *Let V be an arbitrary set, and $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. If $g_v : \mathbb{Z}^m \to \mathbb{R}$ is a family of probability distributions indexed by all $v \in V$ with the property that*

$$\exists M \in \mathbb{R} \text{ such that } \forall v, Pr[Mg_v(z) \geq f(z); z \overset{\$}{\leftarrow} f] \geq 1 - \epsilon$$

then the distribution of the output of the following algorithm $\mathcal{A}$:

```
$v \overset{\$}{\leftarrow} h$
$z \overset{\$}{\leftarrow} g_v$
output $(z, v)$ with probability $\min\left(\frac{f(z)}{Mg_v(z)}, 1\right)$
```

is within statistical distance ϵ/M of the distribution of the following algorithm $\mathcal{F}$:

```
$v \overset{\$}{\leftarrow} h$
$z \overset{\$}{\leftarrow} f$
output $(z, v)$ with probability $1/M$
```

Moreover, the probability that $\mathcal{A}$ outputs something is at least $(1 - \epsilon)/M$.

Proof. For each $v \in V$, define S_v to be the set that consists of all $z \in \mathbb{Z}^m$ such that $Mg_v(z) \geq f(z)$. Notice that by definition, for all $z \in S_v$, the probability that $\mathcal{A}$ outputs z is $g_v(z) \min\left(\frac{f(z)}{Mg_v(z)}, 1\right) = \frac{f(z)}{M}$ and for all $z \notin S_v$ the probability that z is output is $g_v(z)$. We will now bound the probability that the algorithm $\mathcal{A}$ produces some output.

$$Pr[\mathcal{A}\,\text{outputs something}] = \sum_{v\in V} h(v) \left(\sum_{z \in S_v} \frac{f(z)}{M} + \sum_{z \notin S_v} g_v(z) \right) \geq \sum_{v \in V} h(v) \sum_{z \in S_v} \frac{f(z)}{M} \geq \frac{1-\epsilon}{M},$$

and

$$\begin{aligned} Pr[\mathcal{A}\,\text{outputs something}] &= \sum_{v\in V} h(v) \left(\sum_{z \in S_v} \frac{f(z)}{M} + \sum_{z \notin S_v} g_v(z) \right) \\ &\leq \sum_{v\in V} h(v) \left(\sum_{z \in S_v} \frac{f(z)}{M} + \sum_{z \notin S_v} \frac{f(z)}{M} \right) = \frac{1}{M}. \end{aligned}$$

We now move on to bounding the statistical distance of the distribution of the output of $\mathcal{A}$ and $\mathcal{F}$. Let $N_{\mathcal{A}}$, $N_{\mathcal{F}}$ be the probabilities that $\mathcal{A}$ and $\mathcal{F}$ do not output anything, respectively. It's clear that $N_{\mathcal{F}} = 1 - \frac{1}{M}$, and from above, we know that $1 - \frac{1}{M} \leq N_{\mathcal{A}} \leq 1 - \frac{1-\epsilon}{M}$. Then, we have

$$\begin{aligned} \Delta(\mathcal{A}, \mathcal{F}) &= \frac{1}{2} \left(\sum_{z \in \mathbb{Z}^m,\, v \in V} |\mathcal{A}(z,v) - \mathcal{F}(z,v)| + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &= \frac{1}{2} \left(\sum_{z \in \mathbb{Z}^m} \sum_{v \in V} \left| h(v) g_v(z) \min\left(\frac{f(z)}{Mg_v(z)}, 1 \right) - h(v) \frac{f(z)}{M} \right| + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &\leq \frac{1}{2} \left(\sum_{z \in \mathbb{Z}^m} \sum_{v \in V} h(v) \left| g_v(z) \min\left(\frac{f(z)}{Mg_v(z)}, 1 \right) - \frac{f(z)}{M} \right| + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &= \frac{1}{2} \sum_{v \in V} h(v) \left(\sum_{z \in \mathbb{Z}^m} \left| g_v(z) \min\left(\frac{f(z)}{Mg_v(z)}, 1 \right) - \frac{f(z)}{M} \right| + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &\leq \frac{1}{2} \sum_{v \in V} h(v) \left(\sum_{z \in S_v} \left| \frac{f(z)}{M} - \frac{f(z)}{M} \right| + \sum_{z \notin S_v} \left| g_v(z) - \frac{f(z)}{M} \right| + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &\leq \frac{1}{2} \sum_{v \in V} h(v) \left(\sum_{z \notin S_v} \frac{f(z)}{M} + |N_{\mathcal{A}} - N_{\mathcal{F}}| \right) \\ &\leq \frac{1}{2} \left(\frac{\epsilon}{M} + \left(\left(1 - \frac{1-\epsilon}{M} \right) - \left(1 - \frac{1}{M} \right) \right) \right) = \frac{\epsilon}{M} \end{aligned}$$

□

To complete the proof of Theorem 4.6, we let the set V in Lemma 4.7 be all vectors $\mathbf{v} \in \mathbb{Z}^m$ of length at most T, the function f be D_{σ}^m, and the functions g_v be $D_{\mathbf{v},\sigma}^m$. □

Signing Key: $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
Verification Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{T} \leftarrow \mathbf{AS}$
Random Oracle: $\mathrm{H} : \{0,1\}^* \to \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$

```
Sign($\mu, \mathbf{A}, \mathbf{S}$)
    $\mathbf{y} \xleftarrow{\$} D_\sigma^m$
    $\mathbf{c} \leftarrow \mathrm{H}(\mathbf{Ay}, \mu)$
    $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$
    output $(\mathbf{z}, \mathbf{c})$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$
```

```
Verify($\mu, \mathbf{z}, \mathbf{c}, \mathbf{A}, \mathbf{T}$)
    Accept iff
        $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = \mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu)$
```

Fig. 1. Signature Scheme.

	I	II	III	IV	V
n	512	512	512	512	512
q	2^{27}	2^{25}	2^{33}	2^{18}	2^{26}
d	1	1	31	1	31
k	80	512	512	512	512
η	1.1	1.1	1.2	1.3	1.3
$m \approx 64 + n \cdot \log q / \log(2d+1)$	8786	8139	3253	-	-
$m = 2n$ (used in Section 6)	-	-	-	1024	1024
κ s.t. $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$	28	14	14	14	14
$\sigma \approx 12 \cdot d \cdot \kappa \cdot \sqrt{m}$	31495	15157	300926	-	-
$\sigma \approx 6 \cdot d \cdot \kappa \cdot \sqrt{m}$ (used in Section 6)	-	-	-	2688	83328
$M \approx \exp\left(12d\kappa\sqrt{m}/\sigma + (d\kappa\sqrt{m}/2\sigma)^2\right)$	2.72	2.72	2.72	7.4	7.4
approximate signature size (bits) $\approx m \log(12\sigma)$	163000	142300	73000	14500	19500
approximate secret key size (bits) $\approx m \cdot k \cdot \log(2d+1)$	2^{20}	$2^{22.5}$	2^{23}	$2^{19.5}$	$2^{21.5}$
approximate public key size (bits) $\approx n \cdot k \cdot \log q$	2^{20}	$2^{22.5}$	2^{23}	$2^{22.1}$	$2^{22.7}$

Fig. 2. Signature Scheme Parameters. The parameters in columns I, II, and III are based on the hardness of the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem where for the β in Theorem 5.1. Columns IV and V are based on the hardness of the $\mathrm{SIS}_{q,n,m,d}$ search problem (see Section 6). Furthermore, the parameters in column V are also compatible with the LWE assumption (see Section 6.1). The security level for all the instantiations is for $\delta \approx 1.007$ (see Section 3.2). For the ring-based instantiations in Section 7, the key sizes are smaller by a factor of k.

5 Signature Scheme Based on SIS

In this section we present our main theoretical result – a signature scheme based, in the random oracle model, on the average-case hardness of the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem for $\beta = \tilde{O}(n)$. The scheme is presented in Figure 1 and the definition of its parameters and some sample instantiations are in Figure 2. We will now explain the workings of the scheme and sketch the intuition for its security.

The secret key is an $m \times k$ matrix $\mathbf{S}$ of random integers of absolute value at most d, and the public key consists of a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and another matrix $\mathbf{T} \in \mathbb{Z}_q^{m \times k}$ which is equal to $\mathbf{AS}$. For concreteness, we will consider distributions to be statistically close if they are $\approx 2^{-100}$ apart, and we will also want ≈ 100 bits of security from our cryptographic hash function H, and so we will assume that the output of H is 100 bits.[5]

To sign a message μ, the signer first picks an m-dimensional vector $\mathbf{y}$ from the distribution D_σ^m, for some standard deviation σ, then computes $\mathbf{c} = \mathrm{H}(\mathbf{Ay}, \mu)$, and finally computes $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$

[5] It is generally considered folklore that for obtaining signatures with λ bits of security using the Fiat-Shamir transform, one only needs random oracles that output λ bits (i.e. collision-resistance is not a requirement). While finding collisions in the random oracle does allow the *valid* signer to produce two distinct messages that have the same signature, this does not constitute a break.

(there is no reduction modulo q in this step!). The potential signature which he outputs is $(\mathbf{z},\mathbf{c})$, but he only outputs it with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}^m(\mathbf{z})},1\right)$. If nothing was output, the signer runs the signing algorithm again until some signature is outputted.

The main idea behind this structure of the signing algorithm is to make the distribution of the signature $(\mathbf{z},\mathbf{c})$ *independent* of the secret key $\mathbf{S}$. The target distribution for the $\mathbf{z}$'s that we will be aiming for is D_σ^m, but the elements $\mathbf{z}$ in the signature scheme come from the distribution $D_{\mathbf{v},\sigma}^m$, where $\mathbf{v}=\mathbf{Sc}$. This is where we will apply the rejection sampling theorem, Theorem 4.6, from Section 4 to show that for an appropriately-chosen value of M and σ, the signature algorithm will output something with probability approximately $1/M$ and the statistical distance between its output is statistically close to the distribution in which $\mathbf{z}$ is chosen from D_σ^m.

Once we decoupled the distribution of the signature from the distribution of the secret key, we can use a forger who successfully breaks the signature to solve the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem for $\beta\approx\tilde{O}(\|\mathbf{z}\|)$. The idea is that given an $\mathbf{A}$, one can create a secret key $\mathbf{S}$ and publish the public key $(\mathbf{A},\mathbf{AS})$. Then one can reply to signing queries of the forger by either using the key $\mathbf{S}$, or simply by producing signatures by generating $\mathbf{z}$ from the distribution D_σ^m and programming the random oracle accordingly. In our proof (Lemma 5.4), we choose the latter approach because in Section 6, we will not know a valid secret key, but we would like to be able to still use the the same lemma there. Once we have a way to reply to signing queries, we use the forking lemma [PS00,BN06] to use the forger's valid signatures to recover a short vector $\mathbf{v}$ such that $\mathbf{Av}=0$. One important caveat is that to prove that $\mathbf{v}\neq 0$, there needs to be a second (unknown to us) valid secret key $\mathbf{S}'$ such that $\mathbf{AS}=\mathbf{AS}'$, and the forger cannot know which secret key we know. To satisfy the existence of another secret key requires a particular relationship between n,m, and q (Lemma 5.2), and the indistinguishability of $\mathbf{S}$ and $\mathbf{S}'$ is clearly satisfied because the distribution of the signature is independent of the secret key.

We now discuss the verification procedure. Since we tailored $\mathbf{z}$ to be distributed according to D_σ^m, by Lemma 4.4, we know that with probability at least $1-2^{-100}$, we have $\|\mathbf{z}\|<\eta\sigma\sqrt{m}$. And since $\mathbf{Ay}=\mathbf{Az}-\mathbf{Tc}$, the second part of the verification will accept a valid signature.

Theorem 5.1. *If there is a polynomial-time forger, who makes at most s queries to the signing oracle and h queries to the random oracle H, who breaks the signature in Figure 1 (with the relationship between the parameters as in Figure 2) with probability δ, then there is a polynomial-time algorithm who can solve the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem for $\beta=(2\eta\sigma+2d\kappa)\sqrt{m}=\tilde{O}(dn)$ with probability $\approx\frac{\delta^2}{2(h+s)}$. Moreover, the signing algorithm produces a signature with probability $\approx 1/M$ and the verifying algorithm accepts a signature produced by an honest signer with probability at least $1-2^{-m}$.*

Proof. The theorem is proved in a sequence of two Lemmas. In Lemma 5.3, we show that our signing algorithm can be replaced by the one in Hybrid 2 of Figure 3, and the statistical distance between the two outputs will be at most $\epsilon=s(h+s)\cdot 2^{-n+1}+s\cdot\frac{2^{-100}}{M}$. Since Hybrid 2 produces an output with probability exactly $1/M$, the signing algorithm produces an output with probability at least $(1-\epsilon)/M$. Then in Lemma 5.4, we show that if a forger can produce a forgery with probability δ when when the signing algorithm is replaced by one in Hybrid 2, then we can use him to recover a vector $\mathbf{v}$ such that $\|\mathbf{v}\|\le(2\eta\sigma+2d\kappa)\sqrt{m}$ and $\mathbf{Av}=0$ with probability at least $\left(\frac{1}{2}-2^{-100}\right)\left(\delta-2^{-100}\right)\left(\frac{\delta-2^{-100}}{h+s}-2^{-100}\right)\approx\frac{\delta^2}{2(h+s)}$. □

Hybrid 1

```
Sign($\mu, \mathbf{A}, \mathbf{S}$)
  $\mathbf{y} \xleftarrow{\$} D_\sigma^m$
  $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
  $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$
  with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$,
      output $(\mathbf{z}, \mathbf{c})$
      Program $\mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$
```

Hybrid 2

```
Sign($\mu, \mathbf{A}, \mathbf{S}$)
  $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
  $\mathbf{z} \xleftarrow{\$} D_\sigma^m$
  with probability $1/M$,
      output $(\mathbf{z}, \mathbf{c})$
      Program $\mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$
```

Fig. 3. Signing Hybrids

Lemma 5.2. *For any* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *where* $m > 64 + n \cdot \log q / \log(2d+1)$, *for randomly chosen* $\mathbf{s} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^m$, *with probability* $1 - 2^{-100}$, *there exists another* $\mathbf{s}' \in \{-d, \ldots, 0, \ldots, d\}^m$ *such that* $\mathbf{As} = \mathbf{As}'$.

Proof. Notice that $\mathbf{A}$ can be thought of as a linear transformation whose range has size q^n. This means that there are at most q^n elements $\mathbf{s} \in \{-d, \ldots, 0, \ldots, d\}^m$ that do not collide with any other element in $\{-d, \ldots, 0, \ldots, d\}^m$. Since the set $\{-d, \ldots, 0, \ldots, d\}^m$ consists of $(2d+1)^m$ elements, the probability of randomly selecting a non-colliding element is at most

$$\frac{q^n}{(2d+1)^m} \leq \frac{q^n}{(2d+1)^{64+n\log q/\log(2d+1)}} = \frac{1}{(2d+1)^{64}} < 2^{-100}$$

□

Lemma 5.3. *Let $\mathcal{D}$ be a distinguisher who can query the random oracle H and either the actual signing algorithm in Figure 1 or Hybrid 2 in Figure 3. If he makes h queries to H and s queries to the signing algorithm that he has access to, then for all but a* $e^{-\Omega(n)}$ *fraction of all possible matrices* $\mathbf{A}$, *his advantage of distinguishing the actual signing algorithm from the one in Hybrid 2 is at most* $s(h+s) \cdot 2^{-n+1} + s \cdot \frac{2^{-\omega(\log m)}}{M}$, *or more concretely,* $s(h+s) \cdot 2^{-n+1} + s \cdot \frac{2^{-100}}{M}$.

Proof. We first show that the distinguisher $\mathcal{D}$ has advantage of at most $s(h+s)2^{-n+1}$ of distinguishing between the real signature scheme and Hybrid 1. The only difference between the actual signing algorithm and the algorithm in Hybrid 1 is that in Hybrid 1, the output of the random oracle H is chosen at random from $\{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$ and then programmed as the answer to $\mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathrm{H}(\mathbf{Ay}, \mu)$ without checking whether the value for $(\mathbf{Ay}, \mu)$ was already set. Since $\mathcal{D}$ calls H h times, and the signing algorithm s times, at most $s+h$ values of $(\mathbf{Ay}, \mu)$ will ever be set. We now show that each time the Hybrid 1 procedure is called, the probability of generating a $\mathbf{y}$ such that $\mathbf{Ay}$ is equal to one of the previous values that was queried is at most 2^{-n+1}. With probability at least $1 - e^{-\Omega(n)}$, the matrix $\mathbf{A}$ can be written in "Hermite Normal Form" (see Section 3) as $\mathbf{A} = [\bar{\mathbf{A}}||\mathbf{I}]$. Then, for any $\mathbf{t} \in \mathbb{Z}_q^n$,

$$Pr[\mathbf{Ay} = \mathbf{t}; \mathbf{y} \xleftarrow{\$} D_\sigma^m] = Pr[\mathbf{y}_1 = (\mathbf{t} - \bar{\mathbf{A}}\mathbf{y}_0); \mathbf{y} \xleftarrow{\$} D_\sigma^m] \leq \max_{\mathbf{t}' \in \mathbb{Z}_q^n} Pr[\mathbf{y}_1 = \mathbf{t}'; \mathbf{y}_1 \xleftarrow{\$} D_\sigma^n] \leq 2^{-n},$$

where the last inequality follows from Lemma 4.4. Thus if Hybrid 1 is accessed s times, and the probability of getting a collision each time is at most $(s+h)2^{-n+1}$, the probability that a collision occurs after s queries is at most $s(s+h)2^{-n+1}$.

We next show that the statistical distance between the outputs of Hybrid 1 and Hybrid 2 is at most $\frac{2^{-\omega(\log m)}}{M}$. The proof of this fact is almost a direct consequence of Theorem 4.6. Notice that if both Hybrids simply outputted $(\mathbf{z}, \mathbf{v} = \mathbf{Sc})$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$ for Hybrid 1 and probability $1/M$ for Hybrid 2, then Hybrid 1 exactly plays the role of the algorithm $\mathcal{A}$ in Theorem 4.6 and Hybrid 2 corresponds to $\mathcal{F}$ (where the maximum T in Theorem 4.6 corresponds to $d\kappa\sqrt{m}$). But instead of outputting $\mathbf{v} = \mathbf{Sc}$, the Hybrids output just $\mathbf{c}$. But this does not increase the statistical distance because given $\mathbf{v}$, one can generate $\mathbf{c}$ by picking a random element $\mathbf{c} \in \{\mathbf{w} : \mathbf{w} \in \{-1, 0, 1\}^k, \|\mathbf{w}\|_1 \leq \kappa\}$ such that $\mathbf{Sc} = \mathbf{v}$ (for our choice of parameters in this paper, there will actually be only one possible $\mathbf{c}$, with very high probability), and this will have the exact same distribution as the $\mathbf{c}$ in both Hybrids. And finally, since the signing oracle is called s times, the statistical distance is no more than $s \cdot \frac{2^{-\omega(\log m)}}{M}$, or more concretely, $s \cdot \frac{2^{-100}}{M}$, and we obtain the claim in the lemma. □

Lemma 5.4. *Suppose there exists a polynomial-time forger $\mathcal{F}$ who makes at most h queries to the signer in Hybrid 2, s queries to the random oracle H, and succeeds in forging with probability δ. Then there exists an algorithm of the same time-complexity as $\mathcal{F}$ that for a given $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$ finds a non-zero $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq (2\eta\sigma + 2d\kappa)\sqrt{m}$ and $\mathbf{Av} = 0$ with probability at least*

$$\left(\frac{1}{2} - 2^{-100}\right)\left(\delta - 2^{-100}\right)\left(\frac{\delta - 2^{-100}}{h+s} - 2^{-100}\right).$$

Proof. Throughout the proof, let $D_{\mathrm{H}} = \{\mathbf{c} : \mathbf{c} \in \{-1, 0, 1\}^k, \|\mathbf{c}\|_1 \leq \kappa\}$ denote the range of the random oracle H. Given an $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we pick $\mathbf{S} \in \{-d, \ldots, 0, \ldots, d\}^{m \times k}$, and then compute and publish the corresponding verification keys $\mathbf{A}, \mathbf{T} = \mathbf{AS}$. Let $t = h + s$ be the bound on the number of times the the random oracle H is called or programmed during $\mathcal{F}$'s attack. A random oracle query can be made by the forger directly, or the random oracle can be programmed by the signing algorithm when the forger asks to see a signature of some message. We then pick random coins ϕ for the forger and ψ for the signer, and we also pick $\mathbf{r}_1, \ldots, \mathbf{r}_t \overset{\$}{\leftarrow} D_{\mathrm{H}}$, which will correspond to the responses of the random oracle. We now consider a subroutine $\mathcal{A}$, which takes as input $(\mathbf{A}, \mathbf{T}, \phi, \psi, \mathbf{r}_1, \ldots, \mathbf{r}_t)$. The subroutine $\mathcal{A}$ initializes $\mathcal{F}$ by giving it the public key $(\mathbf{A}, \mathbf{T})$ and the random coins ϕ, and then proceeds to run $\mathcal{F}$. Whenever $\mathcal{F}$ wants some message signed, $\mathcal{A}$ runs the signing algorithm in Hybrid 2 using the signer's random coins ψ to produce a signature. During signing, the random oracle H will have to be programmed, and the response of H will be first $\mathbf{r}_i$ in the list $(\mathbf{r}_1, \ldots, \mathbf{r}_t)$ that hasn't been used yet. Of course, $\mathcal{A}$ will have to keep a table of all the queries to H, so in case the same query is made twice, it will have to reply with the previously answered $\mathbf{r}_i$. The forger $\mathcal{F}$ can also make queries to the random oracle, in which case the reply will similarly be the first unused $\mathbf{r}_i$ in the list $(\mathbf{r}_1, \ldots, \mathbf{r}_t)$ (unless the query is not being made for the first time). Once $\mathcal{F}$ finishes running and outputs a forgery (with probability δ), our subroutine $\mathcal{A}$ simply outputs $\mathcal{F}$'s output.

With probability δ, $\mathcal{F}$ will output a message μ and its signature $(\mathbf{z}, \mathbf{c})$ such that $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = \mathrm{H}\left((\mathbf{Az} - \mathbf{Tc}), \mu\right)$. Notice that if the random oracle H was not queried or programmed on some input $\mathbf{w} = (\mathbf{Az} - \mathbf{Tc})$, then $\mathcal{F}$ only has a $1/|D_{\mathrm{H}}|$ chance of producing a $\mathbf{c}$ such that $\mathbf{c} = \mathrm{H}\left(\mathbf{w}, \mu\right)$. Thus with probability $1 - 1/|D_{\mathrm{H}}|$, $\mathbf{c}$ must be one of the $\mathbf{r}_i$'s, and so the probability that $\mathcal{F}$ succeeds in a forgery and $\mathbf{c}$ is one of the $\mathbf{r}_i$'s, is at least $\delta - 1/|D_{\mathrm{H}}|$. Let j be such that $\mathbf{c} = \mathbf{r}_j$. There are two possibilities: $\mathbf{r}_j$ was a response to a random oracle query made by $\mathcal{F}$, or it was programmed during signing. We will deal with the latter, simpler case first.

Suppose that the signer programmed the random oracle $\mathrm{H}\left((\mathbf{A}\mathbf{z}' - \mathbf{T}\mathbf{c}), \mu'\right) = \mathbf{c}$ when signing a message μ'. If the forger outputs a valid forgery $(\mathbf{z}, \mathbf{c})$ for some (possibly different) message μ, then we have $\mathrm{H}\left((\mathbf{A}\mathbf{z}' - \mathbf{T}\mathbf{c}), \mu'\right) = \mathrm{H}\left((\mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c}), \mu\right)$. If $\mu \neq \mu'$ or $\mathbf{A}\mathbf{z}' - \mathbf{T}\mathbf{c} \neq \mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c}$, it means that $\mathcal{F}$ found a pre-image of $\mathbf{r}_j$. Therefore, we have $\mu = \mu'$ and $\mathbf{A}\mathbf{z}' - \mathbf{T}\mathbf{c} \neq \mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c}$, and so $\mathbf{A}(\mathbf{z} - \mathbf{z}') = 0$. We know that $\mathbf{z} - \mathbf{z}' \neq 0$ (because otherwise $(\mathbf{z}, \mu)$ is exactly the same as the old signature $(\mathbf{z}', \mu')$), and since $\|\mathbf{z}\|, \|\mathbf{z}'\| \leq \eta\sigma\sqrt{m}$, we have that $\|\mathbf{z} - \mathbf{z}'\| \leq 2\eta\sigma\sqrt{m}$.

We now turn to the case that $\mathbf{r}_j$ was a response to a random oracle query made by $\mathcal{F}$. In this case, we first record the signature $(\mathbf{z}, \mathbf{r}_j)$ of $\mathcal{F}$ on the message μ, and then generate fresh random elements $\mathbf{r}'_j, \ldots, \mathbf{r}'_t \xleftarrow{\$} D_{\mathrm{H}}$. We then run the subroutine $\mathcal{A}$ again with inputs $(\mathbf{A}, \mathbf{T}, \phi, \psi, \mathbf{r}_1, \ldots, \mathbf{r}_{j-1}, \mathbf{r}'_j, \ldots, \mathbf{r}'_t)$. By the General Forking Lemma of Bellare and Neven [BN06, Lemma 1], we obtain that the probability that $\mathbf{r}'_j \neq \mathbf{r}_j$ and the forger uses the random oracle response $\mathbf{r}'_j$ (and the query associated to it) in its forgery is at least

$$\left(\delta - \frac{1}{|D_{\mathrm{H}}|}\right)\left(\frac{\delta - 1/|D_{\mathrm{H}}|}{t} - \frac{1}{|D_{\mathrm{H}}|}\right),$$

and thus with the above probability, $\mathcal{F}$ outputs a signature $(\mathbf{z}', \mathbf{r}'_j)$ of the message μ and $(\mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c}) = (\mathbf{A}\mathbf{z}' - \mathbf{T}\mathbf{c}')$ where we let $\mathbf{c} = \mathbf{r}_j$ and $\mathbf{c}' = \mathbf{r}'_j$. By rearranging terms in the above equality and plugging in $\mathbf{T} = \mathbf{A}\mathbf{S}$, we obtain

$$\mathbf{A}(\mathbf{z} - \mathbf{z}' + \mathbf{S}\mathbf{c}' - \mathbf{S}\mathbf{c}) = 0. \tag{4}$$

Since $\|\mathbf{z}\|, \|\mathbf{z}'\| \leq \eta\sigma\sqrt{m}$, and $\|\mathbf{S}\mathbf{c}\|, \|\mathbf{S}\mathbf{c}'\| \leq d\kappa\sqrt{m}$ we know that $\|\mathbf{z} - \mathbf{z}' + \mathbf{S}\mathbf{c}' - \mathbf{S}\mathbf{c}\| \leq (2\eta\sigma + 2d\kappa)\sqrt{m}$.

Now all we need to show is that $\mathbf{z} - \mathbf{z}' + \mathbf{S}(\mathbf{c}' - \mathbf{c}) \neq 0$. Let i be a position in which $\mathbf{c}_i \neq \mathbf{c}'_i$. By Lemma 5.2, we know there is at least a $1 - 2^{-100}$ chance that there exists another secret key $\mathbf{S}'$ such that all the columns, except for column i, of $\mathbf{S}'$ are the same as $\mathbf{S}$, and $\mathbf{A}\mathbf{S} = \mathbf{A}\mathbf{S}'$. It's clear that with this definition of $\mathbf{S}'$, if $\mathbf{z} - \mathbf{z}' + \mathbf{S}(\mathbf{c}' - \mathbf{c}) = 0$, then $\mathbf{z} - \mathbf{z}' + \mathbf{S}'(\mathbf{c}' - \mathbf{c}) \neq 0$. More generally, this shows that for every distinct key $\mathbf{S}$ such that $\mathbf{z} - \mathbf{z}' + \mathbf{S}(\mathbf{c}' - \mathbf{c}) = 0$, there exists a distinct key $\mathbf{S}'$ which differs from $\mathbf{S}$ only in column i, such that $\mathbf{z} - \mathbf{z}' + \mathbf{S}'(\mathbf{c}' - \mathbf{c}) \neq 0$. And since the subroutine $\mathcal{A}$ does not get these secret keys as input and does not use them for simulating the signing oracle, the forger $\mathcal{F}$ does not know whether we "know" a secret key like $\mathbf{S}$ or like $\mathbf{S}'$, and so we will get a non-zero answer with probability at least 1/2, since each key has an equal probability of being chosen. □

5.1 Setting the Parameters

In Figure 2, we set some sample parameters to demonstrate the influence of their interplay on the sizes of the signature length and the key size. The secret key is an $m \times k$ matrix with coefficients having absolute value at most d, and so it can be represented by $mk \log(2d + 1)$ bits. The public key $\mathbf{A}, \mathbf{T}$ can be spit into two parts – the matrix $\mathbf{A}$ can be shared by all users (and so can be considered as part of the function), whereas the matrix $\mathbf{T}$ is individual. The part of the public key that is individual for each user requires $nk \log q$ bits of storage. The signature size is dominated by the vector $\mathbf{z}$, since $\mathbf{c}$ is just a small bit-string that is the output of the cryptographic hash function H. By design, the vector $\mathbf{z}$ is distributed according to D_σ^m, and by Lemma 4.4, we know that with probability at least $1 - 2^{-100}$, each coefficient of $\mathbf{z}$ is of length at most 12σ. Thus $\mathbf{z}$ can be represented by $m \log(12\sigma)$ bits.

For security, we use the analysis of [GN08,MR08], as discussed in Section 3.2, and assume that the smallest vector $\mathbf{v}$ such that $\mathbf{Av} = 0$ can be produced has the length specified in Equation (3) of that section. We would like this vector $\mathbf{v}$ to have a larger size than the vector that can be extracted from the successful forger, which is given in Lemma 5.4. There are some trade-offs between the sizes of signatures and keys that can be achieved for the same security level. For example, if we change the value of k from 80 in column I to 512 in column II, it has the effect of making the keys larger by a factor of around 6, and at the same time reducing the signature size by a little over 10%. Another interesting trade-off is achieved by raising the value of d as in column III. Notice that what most affects the length of the signature size is the parameter m. By raising the value of d and q, we can lower m, and can reduce the signature size by almost 50% at the expense of slightly increasing the key sizes.

6 Signatures Based on Low-Density SIS and LWE

From the sample instantiations in the previous section, we saw that m is the one parameter that most affects the signature size. In this section we explore the results of breaking the requirement that $m \approx 64 + n \cdot \log q / \log(2d+1)$ (which is required for Lemma 5.2) and show that this still gives us a provably-secure signature scheme (based on the low-density $\mathrm{SIS}_{q,n,m,d}$ problem), but with much smaller signature and key sizes. Let us consider, for example, taking instantiation III in Figure 2 and lowering the value of d from 31 to, say, 1, *without* changing the value of m. The potential advantage of this modification is that the value of σ goes down by a factor of d, which has the effect of making the signature vector $\mathbf{z}$ smaller (by a factor d), which in turn makes it harder for the adversary to produce a forgery, since he now needs to find a vector that is d times smaller than before. This in turn allow us to lower other parameters, such as q and m, which leads to a "virtuous cycle" of reducing the length of the signature.

We now look at what happens to the security proofs if we proceed as described above. The main problem is that Lemma 5.2 is no longer true since for every $\mathbf{T}$, there will now be, with extremely high probability, only one $\mathbf{S}$ for which $\mathbf{AS} = \mathbf{T}$. The fact that there were multiple $\mathbf{S}$'s was crucially used at the end of Lemma 5.4 to argue that a successful forger can be used to extract a small vector $\mathbf{v}$ such that $\mathbf{Av} = 0$. On the other hand, the proof of Lemma 5.3 is not affected by the relationship between d and m, and so the real signature scheme is still indistinguishable from one that uses Hybrid 2 as its signing algorithm. And since Hybrid 2 *does not* use the secret key to produce signatures, for a given $\mathbf{A}$, we can use the secret key $\mathbf{S}$ with small coefficients in the actual signature, but use an $\mathbf{S}'$ with large coefficients (so that there exists an $\mathbf{S}''$ such that $\mathbf{AS}' = \mathbf{AS}''$) in the proof (see Figure 4). If the distribution of the verification key $(\mathbf{A}, \mathbf{AS})$ is computationally indistinguishable from that of $(\mathbf{A}, \mathbf{AS}')$ (and it is, based on the hardness of the low-density $\mathrm{SIS}_{q,n,m,d}$ problem from Definition 3.4), the distinguisher will not be able to tell that he is given an invalid key pair. And since we never use the secret key to provide signatures to the forger in Lemma 5.4, the forger should act in the same way, and we will be able to find a non-zero $\mathbf{v}$ such that $\mathbf{Av} = 0$.

Using the above framework, we can obtain a signature scheme that is based on the hardness of two problems (i.e. both problems need to be hard for our scheme to be secure): the $\mathrm{SIS}_{q,n,m,d}$ decisional problem and the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem with $\beta = (2\eta\sigma + 2d'\kappa)\sqrt{m}$. Thus the optimal parameter settings will be where the two problems are equally hard. For the hardness of the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem we use the bound in Equation (3) and for the hardness of the decisional problem,

```
Hybrid 2
Signing Key: $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
Verification Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{T} \leftarrow \mathbf{AS}$

Sign($\mu, \mathbf{A}, \mathbf{S}$)
  $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
  $\mathbf{z} \xleftarrow{\$} D_\sigma^m$
  with probability $1/M$,
      output $(\mathbf{z}, \mathbf{c})$
      Program H$(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$
```

```
Hybrid 3
Signing Key: $\mathbf{S} \xleftarrow{\$} \{-d', \ldots, 0, \ldots, d'\}^{m \times k}$
Verification Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{T} \leftarrow \mathbf{AS}$

Sign($\mu, \mathbf{A}, \mathbf{S}$)
  $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
  $\mathbf{z} \xleftarrow{\$} D_\sigma^m$
  with probability $1/M$,
      output $(\mathbf{z}, \mathbf{c})$
      Program H$(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$
```

Fig. 4. Key-Generation and Signing Hybrids: d' is set so that $d' = (2\alpha + 1)d + \alpha$ for some positive integer α and $m \geq 64 + n \cdot \log q / \log (2d' + 1)$

we use Equation (2) in Section 3. We formalize the above intuition in two lemmas analogous to Lemmas 5.3 and 5.4 from Section 5.

Lemma 6.1. *Let $\mathcal{D}$ be a distinguisher who can query the random oracle H and either the actual key-generation/signing algorithms in Figure 1 or those in Hybrid 3 in Figure 4. If he makes h queries to H and s queries to the signing algorithm that he has access to, and can distinguish the real world from Hybrid 3 with advantage δ, then he has advantage $\Omega(\delta/k) - \left(s(h+s) \cdot 2^{-n+1} + s \cdot \frac{2^{-\omega(\log m)}}{M}\right)$ in solving the* $\mathrm{SIS}_{q,n,m,d}$ *decision problem.*

Proof. By Lemma 5.3, we know that the statistical distance between Hybrid 2 and the actual signing algorithm is $\left(s(h+s) \cdot 2^{-n+1} + s \cdot \frac{2^{-\omega(\log m)}}{M}\right)$. The only difference between Hybrids 2 and 3 is the manner in which $\mathbf{S}$ is created in the key-generation algorithm. Using Lemma 3.6 and a hybrid argument, we then obtain that distinguishing the verification key in Hybrid 3 from uniform $(\mathbf{A}, \mathbf{U}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times k}$ is as hard as the $\mathrm{SIS}_{q,n,m,d}$ decision problem (with a loss of a factor k in the advantage due to the hybrid argument). And since the verification key in Hybrid 2 is also indistinguishable from uniform based on the harness of the $\mathrm{SIS}_{q,n,m,d}$ decision problem, the two verification keys are computationally indistinguishable. And because the signing algorithm is independent of the key-generation in both hybrids, the claim in the lemma follows. □

Lemma 6.2. *Suppose there exists a polynomial-time forger $\mathcal{F}$ who is given the verification key and access to the signing algorithm from Hybrid 3, and makes at most h queries to the signing algorithm, s queries to the random oracle H, and succeeds in forging with probability δ. Then there exists an algorithm of the same time-complexity as $\mathcal{F}$ that for a given $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ finds a $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq (2\eta\sigma + 2d'\kappa)\sqrt{m}$ and $\mathbf{Av} = 0$ with probability at least*

$$\left(\frac{1}{2} - 2^{-100}\right)\left(\delta - 2^{-100}\right)\left(\frac{\delta - 2^{-100}}{h+s} - 2^{-100}\right).$$

Proof. The proof is exactly the same as the one of Lemma 5.4, with d' playing the role of d. □

6.1 The LWE Problem

In the Learning With Errors (LWE) problem, one is given an oracle that produces ordered pairs of the form $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}$ where the $\mathbf{a}_i$ are uniformly random in $\mathbb{Z}_q^n$, and $b_i = \mathbf{a}_i \cdot \mathbf{s} + e_i$ where $\mathbf{s}$ is some secret vector in $\mathbb{Z}_q^n$ and e_i is some "error" of small absolute value. Regev [Reg09] showed that there is a quantum reduction from approximating SIVP in all lattices to solving random instances of LWE when the errors e_i come from the discrete Normal distribution D_ψ, and Peikert later showed a classical reduction to LWE from some different lattice problems [Pei09].

An equivalent version of LWE, as shown in [ACPS09], is if the secret key is selected from the distribution D_ψ^n rather than from the uniform distribution. In addition, Regev also showed that the decisional version of the LWE problem, where one is asked to decide whether the ordered pairs $(\mathbf{a}_i, b_i)$ come from the uniform distribution or whether they are generated such that $b_i = \mathbf{a}_i \cdot \mathbf{s} + e_i$, is as hard as the search version.

Using the above definitions, observe that if we have a matrix $\mathbf{A} = [\bar{\mathbf{A}}||\mathbf{I}] \in \mathbb{Z}_q^{n\times 2n}$, where $\bar{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times n}$, then distinguishing pairs $(\mathbf{A}, \mathbf{As})$, where each $\mathbf{s} \overset{\$}{\leftarrow} D_\psi^{2n}$, from uniformly distributed pairs in $\mathbb{Z}_q^{n\times 2n} \times \mathbb{Z}_q^{2n}$ is exactly the decisional LWE problem. By the hybrid argument, distinguishing $(\mathbf{A}, \mathbf{AS})$, where each column of the k columns of $\mathbf{S}$ is distributed according to D_ψ^{2n}, from uniformly distributed pairs in $\mathbb{Z}_q^{n\times 2n} \times \mathbb{Z}_q^{2n\times k}$ is also as hard as LWE. Therefore, except for the distribution of the secret key $\mathbf{S}$, the LWE problem is exactly the low-density $\mathrm{SIS}_{q,n,2n,d}$ problem, and so we can easily change the scheme in the previous section based on the hardness of low-density SIS to be based on LWE instead.

The most important feature of the secret key $\mathbf{S}$ that is used in the proofs is the norm of each of its columns. If the norm of $\mathbf{s} \overset{\$}{\leftarrow} D_\psi^m$ is approximately the same as that of a vector $\mathbf{s}' \overset{\$}{\leftarrow} \{-d, \ldots, 0, \ldots, d\}^m$, then the security and correctness of the scheme from this section will go through almost entirely unchanged. It can be seen that if $\psi \approx \sqrt{\frac{d\cdot(d+1)}{3}}$, then the length of $\mathbf{s}$ is approximately the same as that of a vector $\mathbf{s}'$ (since $\|\mathbf{s}\|$ is tightly concentrated around $\psi\sqrt{m}$ and $\|\mathbf{s}'\|$ around $\sqrt{d(d+1)m/3}$). So a scheme based on LWE where $\psi \approx 18$ would have approximately the same signature size and key lengths as the scheme in column V of Figure 2 where $d = 31$.

Notice that the LWE-based scheme in column V produces signatures that are slightly longer than those produced by the scheme in column IV that is based on the $\mathrm{SIS}_{q,n,2n,1}$ problem. At this point, we are not aware of any algorithms that specifically attack $\mathrm{SIS}_{q,n,2n,1}$ which would justify making the signature longer just so that it is based on the hardness of the LWE problem. But in view of the recent algorithm of Arora and Ge [AG11], which uses algebraic attacks to attack the LWE problem with very small errors, there may be reasons to think that the instantiation in column V could be more secure because it uses larger coefficients.

7 Ring Variants of the Signature Scheme

In general, cryptographic schemes based on the SIS and the LWE problems tend to have very large keys sizes, as can be seen in our table in Figure 2. In our case, the reason for this is that the matrices $\mathbf{S}$ and $\mathbf{T}$ have rather large dimensions and every entry in the matrix is independent of the others. A way to reduce the key sizes is to make all the matrices not independent. Consider constructing the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ as follows: pick its first column $\mathbf{a}_0$ uniformly at random from $\mathbb{Z}_q^n$ and then let the the next $n-1$ columns, $\mathbf{a}_1, \ldots, \mathbf{a}_{n-1}$ be the coefficient representation of the polynomial $\mathbf{a}_0\mathbf{x}^i$ in the ring $\mathbb{Z}_q[\mathbf{x}]/\langle\mathbf{f}\rangle$ for some univariate polynomial $\mathbf{f}(\mathbf{x})$ of degree n. The $n+1^{st}$ column of $\mathbf{A}$ is then picked at random, and the next $n-1$ columns are filled in the same fashion as above (for simplicity, assume that m is an integer multiple of n). Notice that with this construction of $\mathbf{A}$, $\mathbf{As}$ is equivalent to polynomial multiplications and additions in the ring $\mathbb{Z}_q[\mathbf{x}]/\langle\mathbf{f}\rangle$.

It was shown by Micciancio [Mic07] that if $\mathbf{f}(\mathbf{x}) = \mathbf{x}^n - 1$, then the function $\mathbf{As}$ is a one-way function based on the worst-case hardness of the shortest vector problem in cyclic lattices, and this result was subsequently improved to show that if $\mathbf{f}(\mathbf{x})$ is an irreducible polynomial over the integers, then the ring equivalent of the ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ problem is as hard as worst-case ideal lattice problems [PR06,LM06,Lyu08b]. A useful irreducible polynomial that has since been used in ring constructions is $\mathbf{x}^n + 1$, where n is a power of 2. We now describe the ring-version of our signature scheme. All the parameters are the same as in Figure 2 (except the parameter k no longer exists), and we assume that $m = \gamma n$ is an integer multiple of n.

The secret key is $\mathbf{s}_1, \ldots, \mathbf{s}_\gamma \in \mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ where every coefficient of every $\mathbf{s}_i$ is chosen uniformly and independently from $\{-d, \ldots, 0, \ldots, d\}$. The public key is then $(\mathbf{a}_1, \ldots, \mathbf{a}_\gamma, \mathbf{t})$ where each $\mathbf{a}_i$ is uniformly random in $\mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ and $\mathbf{t} = \mathbf{a}_1\mathbf{s}_1 + \ldots + \mathbf{a}_\gamma \mathbf{s}_\gamma$. To sign a message μ, we generate $\mathbf{y}_1, \ldots, \mathbf{y}_\gamma \overset{\$}{\leftarrow} D_\sigma^n$ and compute $\mathbf{c} = \mathrm{H}(\mathbf{a}_1\mathbf{y}_1 + \ldots + \mathbf{a}_\gamma \mathbf{y}_\gamma, \mu)$. We then compute $\mathbf{z}_1 \leftarrow \mathbf{s}_1\mathbf{c}_1 + \mathbf{y}_1, \ldots, \mathbf{z}_\gamma \leftarrow \mathbf{s}_\gamma \mathbf{c}_\gamma + \mathbf{y}_\gamma$, and output the signature $(\mathbf{z}_1, \ldots, \mathbf{z}_\gamma, \mathbf{c})$ with probability $\min\left(\frac{D_\sigma^m(\bar{\mathbf{z}})}{M D_{\bar{\mathbf{v}},\sigma}^m(\bar{\mathbf{z}})}, 1\right)$ where $\bar{\mathbf{z}} = [\mathbf{z}_1^T || \ldots || \mathbf{z}_\gamma^T]^T$ and $\bar{\mathbf{v}} = [(\mathbf{s}_1\mathbf{c})^T || \ldots || (\mathbf{s}_\gamma \mathbf{c})^T]^T$. The verification procedure checks that $\|\bar{\mathbf{z}}\| \le 2\sigma\sqrt{m}$ and that $\mathbf{c} = \mathrm{H}(\mathbf{a}_1\mathbf{z}_1 + \ldots + \mathbf{a}_\gamma \mathbf{z}_\gamma - \mathbf{tc}, \mu)$.

The proof that the above scheme is based on the hardness of the ring version of ℓ_2-$\mathrm{SIS}_{q,n,m,\beta}$ is essentially the same as the proof presented in Section 5. Notice that the signature size of the ring scheme stays the same, but the key sizes all get reduced by a factor of k.

For the case of the schemes in Section 6, things are almost the same as well. One theoretical caveat is that there is no ring-equivalent of Theorem 3.5 which shows the equality between the search and decision versions of the ring-$\mathrm{SIS}_{q,n,m,d}$ problems. Thus the ring-version of this construction is only based on the decision version of ring-$\mathrm{SIS}_{q,n,m,d}$. On the other hand, Lyubashevsky, Peikert, and Regev [LPR10] showed that there is a quantum reduction from worst-case lattice problems to the decision version of the ring-LWE problem (for certain rings, of which $\mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ is one when $q = 1 (\bmod 2n)$), and so the ring version of the construction from Section 6.1 is also based on worst-case instances of problems in ideal lattices.

Acknowledgements. I am very grateful to Oded Regev for all his great ideas and suggestions that greatly improved this work. I would also like to thank Daniele Micciancio and Chris Peikert for illuminating discussions about the results of [MP12]. In particular, I would like to thank Daniele for pointing out an error in an earlier version of the proof of Lemma 4.5. I also thank Leo Ducas and Alain Durmus for useful conversations. And finally, I thank the anonymous Eurocrypt reviewers for their valuable comments and suggestions.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[AFFP11] Martin R. Albrecht, Pooya Farshim, Jean-Charles Faugre, and Ludovic Perret. Polly cracker, revisited. In *ASIACRYPT*, 2011.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415, 2011.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP*, pages 1–9, 1999.

[AP11] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625635, 1993.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM Conference on Computer and Communications Security*, pages 390–399, 2006.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography*, pages 499–517, 2010.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[CN11] Yuanmi Chen and Phong Q. Nguyen. Bkz 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *CRYPTO*, pages 116–137, 2010.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO*, pages 112–131, 1997.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51, 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[GS02] Craig Gentry and Michael Szydlo. Cryptanalysis of the revised ntru signature scheme. In *EUROCRYPT*, pages 299–320, 2002.

[HHGP$^+$03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. In *CT-RSA*, pages 122–140, 2003.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.

[HPS01] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NSS: An NTRU lattice-based signature scheme. In *EUROCRYPT*, pages 211–228, 2001.

[IN96] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. *J. Cryptology*, 9(4):199–216, 1996.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, pages 372–389, 2008.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155, 2006.

[LM08] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, pages 37–54, 2008.

[LMPR08] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, pages 54–72, 2008.

[LO83] J. C. Lagarias and Andrew M. Odlyzko. Solving low-density subset sum problems. In *FOCS*, pages 1–10, 1983.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA*, pages 319–339, 2011.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

[Lyu08a] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, pages 162–179, 2008.

[Lyu08b] Vadim Lyubashevsky. *Towards Practical Lattice-Based Cryptography*. PhD thesis, University of California, San Diego, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, pages 465–484, 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012. Full version at `http://eprint.iacr.org/2011/501`.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[MR08] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Chapter in Post-quantum Cryptography*, pages 147–191. Springer, 2008.

[NR09] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptology*, 22(2):139–160, 2009.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In *CRYPTO*, pages 80–97, 2010.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166, 2006.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[RS10] Markus Rückert and Michael Schneider. Estimating the security of lattice-based cryptosystems. Cryptology ePrint Archive, Report 2010/137, 2010. http://eprint.iacr.org/.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47, 2011.